



## BÁO CÁO THỰC HÀNH LAB 2

Thực hành môn Phương pháp học máy trong an toàn thông tin

**Nhóm: X**



### 1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT101.M11.ANTN.1

STT	Họ và tên	MSSV	Email
1	Trần Hoàng Khang	19521671	19521671@gm.uit.edu.vn
2	Nguyễn Tú Ngọc	20521665	20521665@gm.uit.edu.vn

### 2. NỘI DUNG THỰC HIỆN:

STT	Công việc	Kết quả tự đánh giá
1	Câu hỏi 1	100%
2	Câu hỏi 2	100%
3	Câu hỏi 3	100%
4	Câu hỏi 4	100%
5	Câu hỏi 5	100%
6	Câu hỏi 6	100%
7	Câu hỏi 7	100%
8	Câu hỏi 8	100%

# BÁO CÁO CHI TIẾT

## 1. Sinh viên so sánh kết quả băm với VirusTotal và website Python

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả hash chạy được trong file Notebook:

```
[13] 1 print("MD5 hash value: %s " % md5.hexdigest())
      2 print("SHA256 hash value: %s " % sha256.hexdigest())

MD5 hash value: c3917c08a7fe85db7203da6dcaa99a70
SHA256 hash value: cb580eb7dc55f9198e650f016645023e8b2224cf7d033857d12880b46c5c94ef
```

MD5 hash value: **c3917c08a7fe85db7203da6dcaa99a70**

SHA256 hash value:

**cb580eb7dc55f9198e650f016645023e8b2224cf7d033857d12880b46c5c94ef**

- Đối chiếu hash của MD5 với hash kiểm tra trên [trang chủ Python](#):

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		729e36388ae9a832b01cf9138921b383	25007016	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		3e7035d272680f80e3ce4e8eb492d580	18726176	<a href="#">SIG</a>
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.9 and later (updated for macOS 12 Monterey)	8575cc983035ea2f0414e25ce0289ab8	39735213	<a href="#">SIG</a>
<a href="#">Windows embeddable package (32-bit)</a>	Windows		dc9d1abc644dd78f5e48edae38c7bc6b	7521592	<a href="#">SIG</a>
<a href="#">Windows embeddable package (64-bit)</a>	Windows		340408540eeff359d5eaf93139ab90fd	8474319	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		9d7b80c1c23cfb2cccd63ac4fac9766e	9559706	<a href="#">SIG</a>
<a href="#">Windows installer (32-bit)</a>	Windows		133aa48145032e341ad2a000cd3bff50	27194856	<a href="#">SIG</a>
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	c3917c08a7fe85db7203da6dcaa99a70	28315928	<a href="#">SIG</a>

➔ Mã hash trùng nhau, file tải về toàn vẹn và khớp data.

- Đối chiếu hash của SHA256 với hash kiểm tra trên [VirusTotal](#):

0

/ 70

Community Score

✓ No security vendors and no sandboxes flagged this file as malicious

cb580eb7dc55f9198e650f016645023e8b2224cf7d033857d12880b46c5c94ef

python-3.10.0-amd64.exe

27.00 MB

Size

2022-10-03 06:59:29 UTC

3 minutes ago

EXE

direct-cpu-clock-access invalid-rich-pe-linker-version overlay peexe runtime-modules signed

➔ Mã hash trùng nhau, file tải về toàn vẹn và khớp data.

## 2. Sinh viên cho biết quả của đoạn code trên

## &lt;Xem kết quả chi tiết tại file Notebook (.ipynb)&gt;

Kết quả trả về như đúng mô tả, bao gồm:

- Tên các thư viện imports
- Số lượng Sections
- Tên các sections

```
Imports: [['api-ms-win-crt-runtime-l1-1-0', 'api-ms-win-crt-string-l1-1-0', 'api-ms-win-crt-private-l1-1-0',  
The number of sections: [6, 3]  
Sections: [['.text', '.rdata', '.data', '.pdata', '.didat', '.reloc'], ['.text', '.rsrc', '.reloc']]
```

3. Sinh viên tự tìm hiểu, cài đặt (<https://cuckoo.sh/docs/introduction/index.html>), thực hiện và trình bày phân tích động một tập tin PE.

**Cài đặt Cuckoo Sandbox trên localhost (Ubuntu 20.04 LTS):**

**Lưu ý:** Khi cài đặt cuckoo, đảm bảo có **một máy Host** (tức là một máy chính: có thể dùng VirtualBox, VMWare, ...) và **một máy Guest** (Đây là môi trường Sandbox, để con Cuckoo bỏ file vào chạy thử, và môi trường này phải tắt hết tính năng bảo vệ cơ bản đi, thường là dùng WindowXP hoặc Window7 nằm trong một máy ảo của máy ảo trên).



➔ Cài đặt theo link sau (đảm bảo 100% được nhưng phải làm đúng phiên bản, kỹ và y chang theo vid):

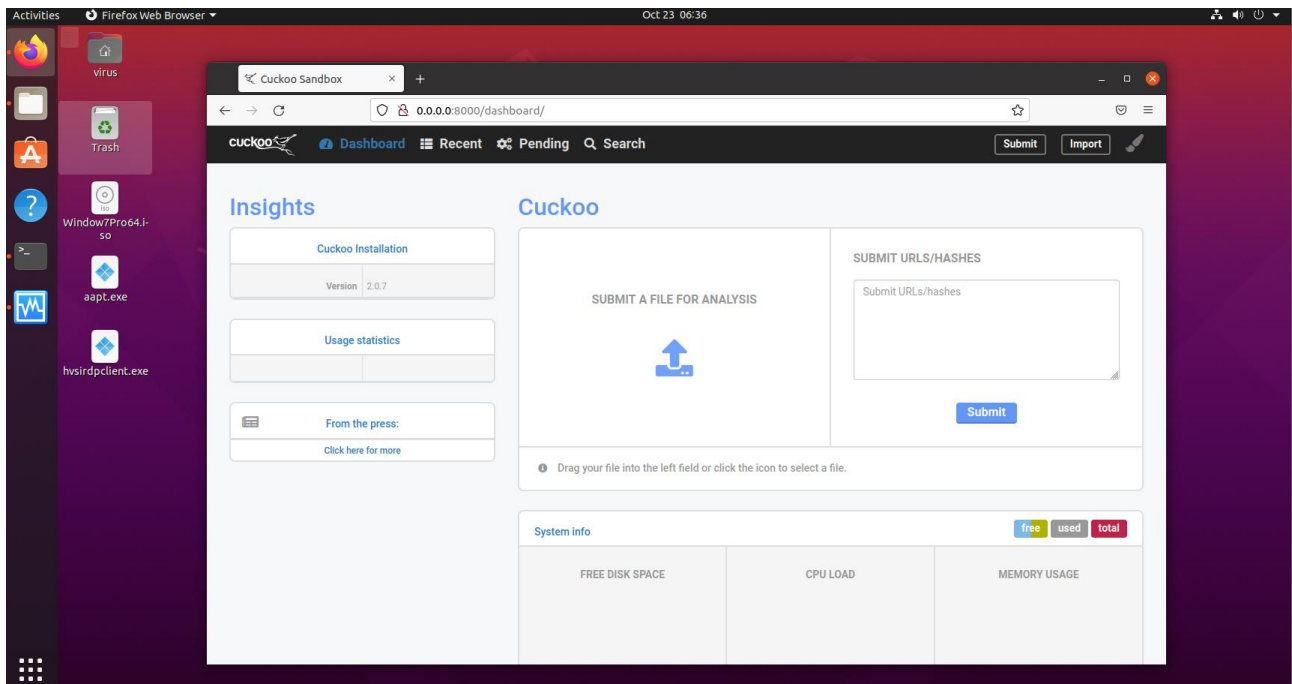
[https://www.youtube.com/watch?v=sWGmtTlzc60&ab\\_channel=ForeGuards](https://www.youtube.com/watch?v=sWGmtTlzc60&ab_channel=ForeGuards)

Link Github gốc: (để cài nhanh không cần động tới vid)

<https://github.com/ForeGuards/Cuckoo-Installation-Guide/blob/main/installation.txt#L186>

**Lưu ý:** Thêm một bước quan trọng trong khi cài (Video không nói). Tắt hẳn "Window Firewall" trên Window 7 trước khi lưu Snapshot: Snapshot1

Sau khi cài xong, ta có giao diện web:

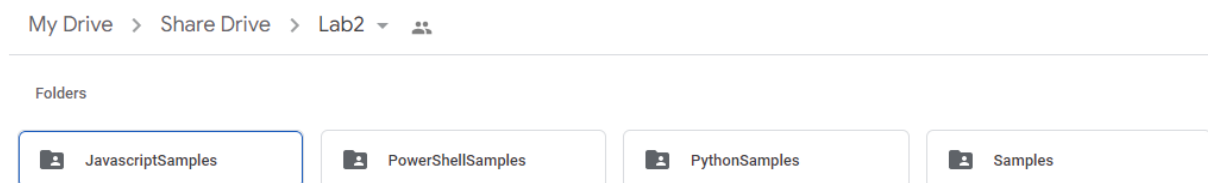


Phần demo trích xuất feature file còn lại, hãy xem video của mình thao tác trên [https://youtu.be/jw74rjz\\_q80](https://youtu.be/jw74rjz_q80)

#### 4. Tương tự sinh viên hãy làm các câu truy vấn về Python và Powershell

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả trích xuất vào 3 folder như trong code:



**Note:** Folder "Sample" không liên quan.

#### 5. Sinh viên cho biết quả của đoạn code trên

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả đạt 95.8%

```
0.958041958041958
[[ 48  2  2]
 [  0 63  1]
 [  0  7 163]]
```

## 6. Sinh viên cho biết quả của đoạn code trên

File đầu tiên test hash và độ giống nhau của 4 string

```
virus@ubuntu:~/Desktop$ python3 firstCheck.py
3:f4oo8MRwRJFGW1gC6uWvPMFSL+JuBF8BSnJi:f4kPvtHM0byFtQ
3:f4oo8MRwRJFGW1gC6uWvPMFSL+JuBF8BS+EFECJi:f4kPvtHM0byFIsJQ
3:f4oo8MRwRJFGW1gC6uWvPMFSL+JuBF8BS6:f4kPvtHM0byF0
3:60QKZ+4CDTfDaRFKYLVL:ywKDC2mVL
100
36
34
0
```

- Ở 4 dòng đầu, ta thấy 4 file cho ra mã hash hoàn toàn khác nhau vì nội dung khác biệt nhau. Ở 4 dòng sau, so sánh hash1-hash1 (giống nhau hoàn toàn), hash1-hash2 và hash1-hash3(gần giống) và hash1-hash4(hoàn toàn khác).

Tiếp theo, ta có một file “Cài đặt python” **gốc** và một file **fake** được thêm 1 byte vào cuối file bằng lệnh “truncate”.

```
virus@ubuntu:~/Desktop$ hexdump -C python-3.10.0-amd64.exe | tail -5
01b010e0 10 9c 34 66 02 d3 51 8c b1 64 19 f3 55 12 0e 74 |..4f..Q..d..U..t|
01b010f0 38 71 4c 2e 1c db 44 d4 f3 81 31 a5 9c 2e c6 06 |8qL...D...1.....|
01b01100 4f 33 c6 8a 9a 5e 16 52 8c 4b 55 10 2b cd 45 61 |03...^..R.KU..+.Ea|
01b01110 a5 00 00 00 00 00 00 00 |.....|
01b01118
virus@ubuntu:~/Desktop$ hexdump -C python-3.10.0-amd64-fake.exe | tail -5
01b010e0 10 9c 34 66 02 d3 51 8c b1 64 19 f3 55 12 0e 74 |..4f..Q..d..U..t|
01b010f0 38 71 4c 2e 1c db 44 d4 f3 81 31 a5 9c 2e c6 06 |8qL...D...1.....|
01b01100 4f 33 c6 8a 9a 5e 16 52 8c 4b 55 10 2b cd 45 61 |03...^..R.KU..+.Ea|
01b01110 a5 00 00 00 00 00 00 00 |.....|
01b01119
```

Chạy code check độ giống nhau tương tự như khi so sánh với chuỗi ta có kết quả được xem là **giống nhau 100%**

```
virus@ubuntu:~/Desktop$ python3 newCheck.py
100
```

Vậy thêm 1 byte hoàn toàn trick được SSDeep.

## 7. Sinh viên cho biết quả của đoạn code trên

<Xem kết quả chi tiết tại file Notebook (.ipynb)>

Kết quả khi lấy N-grams với từng phương pháp:

```
[ ] 1 # Frequency
    2 import numpy as np
    3 X = np.asarray(X)
    4 X_top_K2_freq = X[:, :K2]
    5 X_top_K2_freq

array([[10935, 4673, 7, ..., 13, 248, 180],
       [15237, 2604, 866, ..., 630, 17, 507],
       [ 4963, 282, 88, ..., 120, 0, 37],
       ...,
       [11338, 2286, 9, ..., 3, 255, 148],
       [35587, 196, 5, ..., 2, 1, 3600],
       [ 3826, 71, 14, ..., 14, 13, 110]])

[ ] 1 # Mutual information algorithm
    2 mi_selector = SelectKBest(mutual_info_classif, k=K2)
    3 X_top_K2_mi = mi_selector.fit_transform(X, y)
    4 X_top_K2_mi

array([[320, 0, 39, ..., 3, 0, 0],
       [100, 133, 119, ..., 0, 22, 2],
       [ 5, 6, 2, ..., 0, 2, 0],
       ...,
       [315, 2, 51, ..., 2, 0, 0],
       [942, 11, 248, ..., 6, 7, 9],
       [760, 11, 33, ..., 15, 11, 15]])

[ ] 1 # Chi-squared algorithm
    2 chi2_selector = SelectKBest(chi2, k=K2)
    3 X_top_K2_ch2 = chi2_selector.fit_transform(X, y)
    4 X_top_K2_ch2

array([[10935, 4673, 7, ..., 79, 34, 15],
       [15237, 2604, 866, ..., 2, 19, 3],
       [ 4963, 282, 88, ..., 1, 0, 0],
       ...,
       [11338, 2286, 9, ..., 29, 23, 8],
       [35587, 196, 5, ..., 38, 150, 22],
       [ 3826, 71, 14, ..., 12, 423, 7]])
```

## 8. Sinh viên hoàn thành các bước trên

<Xem kết quả tại file Notebook (.ipynb)>



<Xem các bước chi tiết tại file Notebook (.ipynb)>

Đã comment giải thích